Better approximation guarantees for the NSGA-II by using the current crowding distance

Weijie Zheng, Benjamin Doerr

Better Approximation Guarantees for the NSGA-II by Using the Current Crowding Distance

Weijie Zheng
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China

Benjamin Doerr*
Laboratoire d'Informatique (LIX)
École Polytechnique, CNRS
Institut Polytechnique de Paris
Palaiseau, France

Abstract

A recent runtime analysis (Zheng, Liu, Doerr (2022)) has shown that a variant of the NSGA-II algorithm can efficiently compute the full Pareto front of the OneMinMax problem when the population size is by a constant factor larger than the Pareto front, but that this is not possible when the population size is only equal to the Pareto front size. In this work, we analyze how well the NSGA-II approximates the Pareto front when it cannot compute the whole front. We observe experimentally and by mathematical means that already when the population size is half the Pareto front size, relatively large gaps in the Pareto front remain. The reason for this phenomenon is that the NSGA-II in the selection stage computes the crowding distance once and then repeatedly removes individuals with smallest crowding distance without updating the crowding distance after each removal. We propose an efficient way to implement the NSGA-II using the momentary crowding distance. In our experiments, this algorithm approximates the Pareto front much better than the previous version. We also prove that the gaps in the Pareto front are at most a constant factor larger than the theoretical minimum.

*Corresponding author.

1 Introduction

While the theory of evolutionary algorithms (EAs), in particular, the mathematical runtime analysis, has made substantial progress in the last 25 years [NW10, AD11, Jan13, DN20], the rigorous understanding of multi-objective EAs (MOEAs) is less developed and is massively lagging behind their success in practice. However, the last few years saw significant activity, for example [BQT18, RNNF19, QYT+19, ZYQ19, QBF20, BFQY20, DZ21, Cra21]. In particular, now the first analyses of MOEAs that are used in practice (as opposed to synthetic example) have appeared, namely several analyses of the MOEA/D [LZZZ16, HZCH19, HZ20] and a first analysis of the NSGA-II [ZLD22], the by far dominant algorithm in practice [ZQL+11].

The analysis of the NSGA-II [ZLD22] proved that several variants of this algorithm can compute the full Pareto front of the OneMinMax benchmark efficiently when the population size is chosen by a constant factor larger than the size of the Pareto front (which is $n+1$ for the OneMinMax problem). However, it was also proven that a population size larger than the Pareto front is necessary – if these two sizes are only equal, then with probability $1 - \exp(-\Omega(n))$ for an exponential number of iterations the population of the NSGA-II will not cover a constant fraction of the Pareto front. Experiments show that this fraction is roughly 20% for the OneMinMax benchmark and roughly 40% for the LOTZ benchmark.

Since we cannot assume that the NSGA-II is usually run with a population size larger than the Pareto front by a constant factor – both because the algorithm user does not know the size of the Pareto front and because some problems have a so large Pareto front that using a comparably large population size is not possible –, a deeper understanding of the performance of the NSGA-II in such situations is highly desirable (and this is our target in this work).

There is some reason to be optimistic: The experiments conducted in [ZLD22] for the case that the population size equals the size of the Pareto front not only showed that 20% or 40% of the Pareto front was not covered, but they also show that the missing points are relatively evenly distributed over the Pareto front (the largest empty interval ever seen was of length 4). Hence the population evolved by the NSGA-II in these experiments was a good approximation of the Pareto front.

Our more detailed analysis of the approximation qualities of the NSGA-II cannot fully support this optimism, but still shows a fair approximation capability of the NSGA-II. More importantly, we detect in the selection mechanism of the NSGA-II a reason for the lower-than-expected approximation capability, we propose a modification of the algorithm, and we prove that this modified NSGA-II approximates well the Pareto front.

In detail, when we ran the experiments for the OneMinMax with problem size $n = 601$ and population sizes $N = (n+1)/2 = 301, \lceil (n+1)/4 \rceil = 151$,

and $\lceil (n+1)/8 \rceil = 76$, empty interval with size at least twice $\lceil \frac{(n+1)-1}{N-1} \rceil$ remains. Unfortunately, due to the complicated population dynamics of the NSGA-II, we were not able to support these experimental findings with a mathematical analysis. Nevertheless, the experimental results are quite clear and show that even on a simple benchmark like OneMinMax, the largest empty interval the population has on the Pareto front is much larger than the ideal value $\frac{(n+1)-1}{N-1}$, which would result from a perfect distribution of the population on the Pareto front.

To better understand how this discrepancy can arise, we regard two synthetic examples. We show that when the combined parent and offspring population is such that each point on the Pareto front is covered exactly once (this implies that the population size is essentially half the size of the Pareto front), then with high probability the next parent population does not cover an interval of length $\Theta(\log n)$ on the Pareto front (whereas simply removing every second point would give a population such that each point on the Pareto front has a neighbor that is covered by the population). We further construct a more artificial example where the combined parent and offspring population covers the Pareto front apart from isolated points, but the next parent population does not cover an interval of length $n/3$ of the Pareto front.

The reason why we were able to construct such examples is the following property of the selection scheme of the NSGA-II. To select the new parent population, the NSGA-II uses as first criterion the non-dominated sorting and then the crowding distance. The crowding distance, however is not updated during the selection process. That is, while removing individuals with smallest crowding distance, the changing crowding distance of the remaining individuals is not taken into account, but instead the algorithm proceeds with the initial crowding distance. We assume that this design choice was made for reasons of efficiency – by not updating the crowding distance, it suffices to sort the combined parent and offspring population and then remove the least interesting half of the individuals.

Since, as our examples show, this procedure can lead to very imbalanced selections, we design a version of the NSGA-II which updates the crowding distance after each removal. Consequently, in the selection step, when comparing individuals in the same non-dominated sorting front, it repeatedly removes the individuals that at that moment have the smallest crowding distance. We note that this procedure can be implemented very efficiently: The removal of one individual changes the crowding distance of at most 4 other individuals (in a bi-objective problem), so at most 4 crowding distance values need to be updated. There is no need for a new sorting from scratch when we use as a data structure a priority queue. With this implementation, the whole selection takes not more than $O(N \log N)$ operations, which is the same order of magnitude as the one sorting in the original NSGA-II.

For this modified NSGA-II, the problems shown above cannot occur. We prove that this algorithm for the OneMinMax benchmark computes efficiently a population such that the largest empty interval on the Pareto front has length at most $\frac{4n}{N-3}$, hence at most a constant factor larger than the theoretical minimum of $\frac{(n+1)-1}{N-1}$. Hence when the population size not large enough to cover the full Pareto front, this algorithm computes very good approximations to the Pareto front. Our experiments confirm this finding. For the problem size $n = 601$, the modified algorithm could almost reach the ideal MEI value for all three different population sizes $N = 301, 151$, and 76, while the traditional NSGA-II have at least twice MEI value than the ideal one.

This work is organized as follows. Section 2 briefly introduces the bi-objective optimization and the NSGA-II. Section 3 discusses the approximation assessment that will be used in this work. The difficulty caused by the survival selection in traditional NSGA-II is theoretically shown via two synthetic examples in Section 4, and Section 5 introduces our modified variant and conducts the theoretical analysis on its approximation ability. Our experimental verifications are shown in Section 6, and Section 7 concludes this work.

2 Preliminaries

2.1 Bi-objective Optimization

This paper concentrates on the bi-objective optimization problem $f = (f_1, f_2) : \{0,1\}^n \to \mathbb{R}^2$ with each objective to be maximized. For $x, y \in \{0,1\}^n$, we say x *strictly dominates* y, denoted by $x \succ y$, if $f_1(x) \geq f_1(y), f_2(x) \geq f_2(y)$ and at least one of the inequalities holds. If x cannot be strictly dominated by any solution in $\{0,1\}^n$, we say x is *Pareto optimal* and $f(x)$ a *Pareto front point*. The set of all Pareto front points is called *Pareto front*. The aim for a multi-objective optimizer is to obtain as many points in the Pareto front as possible.

Here we introduce one popular bi-objective benchmarks, OneMinMax, in evolutionary theory community. OneMinMax is first proposed by [GL10] and stems from the early COCZ benchmark [LTZ+02]. The first objective of the OneMinMax counts the number of zeros in the bit-string, and the second objective counts the number of ones. More specifically, for any $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, the OneMinMax function is defined as

$$f = (f_1, f_2) = \left(n - \sum_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i\right).$$

It is not difficult to see that any solution $x \in \{0,1\}^n$ is Pareto optimal, and the Pareto front is $(0, n), (1, n-1), \ldots, (n, 0)$.

Our following theoretical and empirical analyses will be conducted on the OneMinMax to see the approximation ability.

2.2 NSGA-II

We give a brief introduction on the NSGA-II, which is first proposed in [DPAM02], and the by far dominant algorithm in practice [ZQL+11]. It has a fixed population size N. Each time new individuals are generated, MOEAs need to adopt a strategy to remove some individuals to maintain a fixed population size. NSGA-II designs the complete order that uses the dominance as the first criterion and the diversity measure (crowding distance) as the second criterion so that every two individuals in the population are comparable, and removes the worst individuals.

Detailedly, after the random initialization, in each generation, the offspring Q_t with population size of N is generated from the parent population P_t, and then NSGA-II need to remove N individuals from the combined population $R_t = P_t \cup Q_t$. It divides R_t into several fronts $F_1, F_2, \ldots$ where F_1 is the set of the non-dominated solutions in R_t, and $F_i, i > 1$ is the set of the non-dominated solutions in $R_t \setminus \{F_1, \ldots, F_{i-1}\}$. For the first index i^* such that the size of $\cup_{i=1}^{i^*} F_i$ is larger than N, the NSGA-II will calculate the crowding distance, denoted by cDis of the individuals in F_{i^*}. For each objective, it sorts the individuals according to their objective values, the cDis value of the first and last points in the sorted list is infinite, and the cDis of an individual with respect to the current objective is the distance of the objective values of its two neighbors in the list. The complete cDis of the individual is the sum of its cDis components for all objectives. Then $|\cup_{i=1}^{i^*} F_i| - N$ number of individuals in F_{i*} with the smallest crowding distance value will be removed, and the tie is broken randomly. The procedures for the calculation of the crowding distance and the NSGA-II framework is respectively shown in Algorithm 1 and Algorithm 2. Besides, we call it "*fair (mating) selection*" when generating N offspring from the parent population that each parent will generate one offspring, and call it "*random selection* when generating N offspring via N times choosing an individual uniformly at random from the parent population.

Zheng, Liu, and Doerr [ZLD22] recently conducted the first mathematical runtime analysis on the NSGA-II, that is, the first series of theoretical results about the number of fitness evaluations that the NSGA-II needs to cover the Pareto front. They showed that the NSGA-II with several mating selection and mutation strategies can efficiently (with the same runtime complexity as the basic global simple evolutionary multi-objective optimizer (GSEMO) [Gie03], that is, $O(n^2 \log n)$ for the n-dimensional OneMinMax and $O(n^3)$ for the n-dimensional LOTZ) cover the full Pareto front when the population size is a constant factor larger than the Pareto front size of the OneMinMax and LOTZ with the constant properly chosen. However, they also theoretically and empirically showed

Algorithm 1 crowding-distance(S)

Input: $S = \{S_1, \ldots, S_{|S|}\}$: the set of individuals

Output: $\text{cDis}(S) = \Big(\text{cDis}(S_1), \ldots, \text{cDis}\Big(S_{|S|}\Big)\Big)$ where $\text{cDis}(S_i)$ is the crowding distance for S_i

1: $\text{cDis}(S) = 0$
2: **for** each objective f_i **do**
3: Sort S in order of descending f_i value: $S_{i.1}, \ldots, S_{i.|S|}$
4: $\text{cDis}(S_{i.1}) = +\infty, \text{cDis}\Big(S_{i.|S|}\Big) = +\infty$
5: **for** $j = 2, \ldots, |S| - 1$ **do**
6: $\text{cDis}(S_{i.j}) = \text{cDis}(S_{i.j}) + \frac{f_i(S_{i.j-1}) - f_i(S_{i.j+1})}{f_i(S_{i.1}) - f_i\left(S_{i.|S|}\right)}$
7: **end for**
8: **end for**

Algorithm 2 NSGA-II

1: Uniformly at random generate the initial population $P_0 = \{x_1, x_2, \ldots, x_N\}$ for $x_i \in \{0,1\}^n, i = 1, 2, \ldots, N$.
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Generate the offspring population Q_t with size N
4: Using the fast-non-dominated-sort() in [DPAM02] to divide R_t into $F_1, F_2, \ldots$
5: Find $i^* > 1$ such that $|\cup_{i=1}^{i^*-1} F_i| < N$ and $|\cup_{i=1}^{i^*} F_i| \geq N$, or $i^* = 1$ for $|F_1| \geq N$
6: Using the crowding-distance() in Algorithm 1 to separately calculate the crowding distance of each individual in $F_1, \ldots, F_{i^*}$
7: Let $\tilde{F}_{i^*}$ be the $N - |\cup_{i=1}^{i^*-1} F_i|$ individuals in F_{i^*} with largest crowding distance, chosen at random in case of a tie
8: $P_{t+1} = \left(\cup_{i=1}^{i^*-1} F_i\right) \cup \tilde{F}_{i^*}$
9: **end for**

the exponential runtime to not cover the full Pareto front with population size the same as the Pareto front size. As far as we know, [ZLD22] is the only work theoretically discussing the runtime (number of iterations or fitness evaluations) about the NSGA-II, although there is literature about the implementational complexity of the operations for each iteration, like the original NSGA-II paper [DPAM02].

3 Approximation Assessment

Although [ZLD22] pointed the inefficiency of the NSGA-II with not proper population size (see in the previous section), their experiments also indicated the fair approximation ability of the NSGA-II as the maximal empty interval size was of length 4. As this work will focus on the approximation ability, this section will introduce the approximation assessments for the MOEAs.

3.1 Related Assessment

The approximation of the Pareto front is usually considered when the final set of the objectives of the Pareto optima that one MOEA reached is smaller than the full Pareto front for the given problem, or even when the full Pareto front is not known (usually in the practical applications). For the theoretical analysis, the multiplicative ε-dominance relation [LTDZ02] is used in [HN08] and [BFN08] to show that for a large front function the GSEMO cannot have a good approximation in expected polynomial time but the ε-dominance based algorithm and the $(\mu + 1)$-simple indicator-based evolutionary algorithm ($(\mu + 1)$-SIBEA) can efficiently reach such approximation. Although we are aware of the other approximation assessments used in practice for both multi-objective and many-objective optimization [LLTY15], like hypervolume indicator [ZT98], and inverted generational distance [BT03], we just give a brief introduction on the ε-dominance as it is used in the theory community, but we are optimistic (and leave it as our future work) that the assessment discussed in this paper can also be transferred to these indicators since the analyzed problems are relatively simple.

The multiplicative ε-dominance relation generalizes the common "dominance" concept by not requiring the exact comparison of the function values between two solutions but the comparison of the function values of one solution multiplied with $(1 + \varepsilon)$ and the exact function value of the other solution. In this definition, more solutions could be comparable, and the MOEA with good approximation ability should obtain a set of solutions (population) that every Pareto optimal solutions can be ε-dominated by at least one solution in the population. Here is the formal definition. It is not difficult to see that the smaller ε, the better approximation one algorithm could obtain.

Definition 1 (Multiplicative ε-dominance relation [LTDZ02])**.** *Let $\varepsilon > 0$ and $m > 0$ be the number of objectives. For $u, v \in \mathbb{R}^m$, we say u* ε-dominates *v, denoted by $u \succeq_\varepsilon v$, if and only if $(1 + \varepsilon)u \geq v$, that is, $(1 + \varepsilon)u_i \geq v_i$ for all $i = 1, \ldots, m$.*

Let $W = \{u \mid u \in \mathbb{R}^m\}$ be the whole objective vector set for a given problem. We say a subset $S \subset W$ is an ε-approximation *for this problem if and only if for each $v \in W$, there exists $u \in S$ such that $u \succeq_\varepsilon v$.*

3.2 Maximal Empty Interval Size

The ε-dominance is a general measure for the problem with known Pareto front. For the n-dimensional OneMinMax that this paper will analyze, each possible objective value is on the Pareto front and the first objective values of full Pareto front are exactly $0, 1, \ldots, n$. Any missing Pareto front point can be directly seen in $[0..n]$, hence, we now simply use a measure about the size of the maximal empty interval, denoted as MEI, inside $[0..n]$ in terms of the solutions that one MOEA could reach and with respect to f_1 values. If the maximal empty interval size is as small as possible, then the MOEA can approximate the Pareto front as well as possible. We will also show the transformation from the MEI to the general ε-dominance in Section 3.3. The formal definition of the MEI of one set U with respect to one metric h is shown in the following.

Definition 2. *For a given metric $h : \{0,1\}^n \to \mathbb{R}_{\geq 0}$ and a given set $U = \{u_1, \ldots, u_m\} \subset \{0,1\}^n$ with size of m, let $v_1, v_2, \ldots, v_m$ be one sorted list of $h(u_1), \ldots, h(u_m)$ in the order of decreasing value of h. We define the* maximal empty interval size *of U w.r.t. h, denoted by $\mathrm{MEI}(U,h)$, as $\mathrm{MEI}(U,h) = \max\{v_i - v_{i+1} \mid i = 1, \ldots, m-1\}$.*

It is not difficult to see that the ideal (smallest) MEI for OneMinMax and an MOEA with a fixed population size happens when the population is evenly distributed. We explicitly formalize it in the following lemma as it is the tight lower bound of the MEI, and could reflect the discrepancy about how well an MOEA could approximate the Pareto front for the OneMinMax both theoretically and empirically.

Lemma 3. *Consider one MOEA with fixed population size N to optimize the n-dimensional OneMinMax. The ideal (smallest) MEI is $\lceil \frac{n}{N-1} \rceil$.*

3.3 Transformation Between Two Measures

As noted before, for the simple problem OneMinMax that this paper will discuss, our considered MEI approximation measure can be easily transferred to the ε-approximation measure that used in the theory community. The transformation is shown in the following lemma, which can be easily obtained from the definitions. We are also optimistic that for the OneMinMax, MEI can also be transferred to the hypervolume and inverted generational distance. Due to the limited space, we will omit it and leave as our near future work.

Lemma 4. *Let $f = (f_1, f_2)$ be the OneMinMax function defined on bit-strings of length n. Let P be a population of solutions of f. Assume that $0^n, 1^n \in P$ and that $k := \mathrm{MEI}(P, f_1) < \frac{n}{2}$. Then P is an ε-approximation of f for all $\varepsilon \geq \frac{2k-2}{n-2k+2}$.*

Proof. Let $v = (i, n-i)$, $i \in [0..n]$ be any point in the objective space of f. If $v \in f(P)$, then from the definition of ε-dominance, we know $v \succeq_\varepsilon v$ for any $\varepsilon > 0$. Hence, in the following, we just consider the case that $v \notin f(P)$. That is, since $\{0^n, 1^n\} \subset P$, we just consider $i = [1..n-1]$. By symmetry, we may assume that $i \geq \frac{n}{2}$. Let $\varepsilon \geq \frac{2k-2}{n-2k+2}$. We show that there is an element $x \in P$ with $(1+\varepsilon)f(x) \geq v$.

Let $j \in [0..i-1]$ maximal such that there is an $x \in P$ with $f(x) = (j, n-j)$. Since $j < i$, we have $n - j > n - i$ and thus in particular $(1+\varepsilon)(n-j) \geq n-i$. By definition of MEI$(\cdot,\cdot)$, we have $j \geq i - k + 1 \geq \frac{n}{2} - k + 1$. We compute $(1+\varepsilon)j = j + \varepsilon j \geq j + \varepsilon(\frac{n}{2} - k + 1) \geq j + k - 1 \geq i$, the latter again by definition of MEI$(\cdot,\cdot)$. This shows the claim. □

Since our MEI measure is more intuitive and can be easily transferred to the ε-approximation that utilized in the theory community, we will use MEI as the approximation measure to see how close that one MOEA could achieve to the ideal $\lceil \frac{n}{N-1} \rceil$ in the remaining of this work.

4 Difficulties for the NSGA-II to Approximate the Pareto Front

In this section, we show that the traditional way how the NSGA-II selects the next population, namely by relying on the initial crowding distance, can lead to not perfect approximations of the Pareto front. The experiments in Section 6 show that the large gap between the MEI for the traditional NSGA-II and the ideal one, like at least 24 for the first quantile among all data in 20 independent runs while the ideal MEI is 9 for the population size $N = 76$ on the 601-dimensional OneMinMax. In the following, we analyze the result of the selection from two different combined parent and offspring populations. These examples demonstrate quite clearly that the traditional selection can lead to unwanted results. We note that these results do not prove completely that the NSGA-II has difficulties to find good approximations since we do not know how often the NSGA-II enters exactly these situations. Unfortunately the population dynamics of the NSGA-II are too complicated for a full proof. Our experimental results, however, indicate that the phenomena we observe in these synthetic situations (in particular, the first one) do show up.

We start by regarding at the ideal-looking situation that the combined parent and offspring population for each point on the Pareto front contains exactly one individual. Hence by removing the individual corresponding to (essentially) every second point on the Pareto front, one could obtain a very good approximation of the front. Surprisingly, the NSGA-II does much worse. Since all solutions apart

from the two extremal ones have the same crowding distance, the NSGA-II removes a random set of N out of these $2N-2$ inner solutions. As we show now, with high probability this creates an uncovered interval on the Pareto front of length $\Theta(\log n)$. The upper bound for the expectation is proved via calculating of the probability to remove an interval with size $c \log n$ (for a certain large constant c) and a union bound among all possible such intervals as the upper bound of $o(1/n^{c-1})$ for MEI $\geq c \log n$, and then comes the expectation of $O(\log n)$. For the lower bound in expectation, the key is to use the following removal process (with respect to the empty interval size) to deal with the stochastic dependency in the traditional removal process: first selecting N inner individuals with repetition, and then repeating removing random inner individuals until N individuals are removed.

Lemma 5. *Let $n \geq 7$. Consider using the NSGA-II to optimize the n-dimensional* OneMinMax *function. Let the population size $N = (n+1)/2$. Suppose that for a certain generation $t \geq 0$, the combined parent and offspring population R_t fully covers the Pareto front, that is, $f(R_t) = \{(0,n),(1,n-1),\ldots,(n,0)\}$. Then with probability $1-\exp(-n^{2/3}/\ln n)$, we have $\mathrm{MEI}(P_{t+1}, f_1) \geq \lfloor \frac{1}{3} \ln n \rfloor$. On the positive side, $E[\mathrm{MEI}(P_{t+1}, f_1)] = \Theta(\log n)$ and $\Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq c \log_{3/2} n] = n^{1-c}$ for any constant $c > 1$.*

Proof. We note that R_t has size at least $n+1$, since it covers the Pareto front, which has a size of $n+1$. From our assumption $N = (n+1)/2$, we thus conclude that every point in the Pareto front has only one corresponding individual in R_t. Hence, except the individuals 0^n and 1^n that have infinite crowding distance, all other individuals have the equal crowding distance of $4/n$. Consequently, the original NSGA-II survival selection will randomly select N individuals from the $2N-2$ inner individuals (all individuals in R_t except 0^n and 1^n) to be removed.

To prove the logarithmic lower bound on the expectation, we argue as follows. Let $k = \lfloor \frac{1}{3} \ln n \rfloor$. Let $M \subseteq [1..n-k]$ such that $|M| = \lceil n/\ln n \rceil$ and for any two $m_1, m_2 \in M$, we have $|m_1 - m_2| \geq k$ or $m_1 = m_2$ (note that such a set exists since $k|M| \leq n-1$ by definition of k and M). Consequently, for different $m \in M$, the intervals $I_m = [m..m+k-1]$ are disjoint.

To cope with the stochastic dependencies, we regard a particular way to sample the N individuals to be removed. In a first phase, we select N times independently (with repetition) an individual x with $f_1(x) \in [1..n-1]$, recall that these are the ones with smallest crowding distance. This defines a random set of individuals of cardinality at most N (but most probably less than N), and we remove these individuals from the combined parent and offspring population. In a second phase, we repeat removing random individuals x with $f_1(x) \in [1..n-1]$ until we have removed a total of N individuals.

We now prove that already after the first phase, with high probability an interval of length k on the Pareto front is not covered by the population. Apparently, such an interval is also not covered by the final population. Denote by $\tilde{A}_{m,k}$ the probability that all individuals x with $f_1(x) \in I_k$ are removed in the first phase. Then

$$\begin{aligned}\Pr[\tilde{A}_{m,k}] &= \Big(1 - \Big(1 - \tfrac{1}{n-1}\Big)^N\Big)^k \geq \Big(1 - \exp\Big(-\tfrac{N}{n-1}\Big)\Big)^k \\ &\geq \Big(1 - e^{-1/2}\Big)^k \geq (1/e)^k \geq n^{-1/3}.\end{aligned}$$

By construction, the events $\tilde{A}_{m,k}$, $m \in M$, are independent. Hence

$$\Pr[\forall m \in M : \neg\tilde{A}_{m,k}] \leq (1 - n^{-1/3})^{|M|} \leq \exp\Big(-n^{-1/3}\frac{n}{\ln n}\Big) = \exp\Big(-\frac{n^{2/3}}{\ln n}\Big).$$

This proves that with probability at least $1 - \exp(-n^{2/3}/\ln n)$, after the selection phase there is an interval of length k of the Pareto front such that none of its points is covered by the new population P_{t+1}. This also implies $E[\mathrm{MEI}(P_{t+1}, f_1)] \geq (1 - \exp(-n^{2/3}/\ln n))k = \Omega(\log n)$.

We now turn to the upper bounds. Let $k \in [1..N]$ and $m \in [1..n-k]$, and let $A_{k,m}$ be the event that all individuals with f_1 value in $I_m = [m..m+k-1]$ are selected to be removed. Then

$$\Pr[A_{m,k}] = \frac{\binom{2N-2-k}{N-k}}{\binom{2N-2}{N}} = \frac{\frac{(2N-2-k)!}{(N-k)!(N-2)!}}{\frac{(2N-2)!}{N!(N-2)!}} = \frac{N(N-1)\cdots(N-k+1)}{(2N-2)(2N-3)\cdots(2N-1-k)}.$$

It is not difficult to see that if $A_{m,k}$ happens for some m, then $\mathrm{MEI}(P_t, f_1) \geq k$. Hence, by a union bound over all possible m, we obtain

$$\begin{aligned}\Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq k] &\leq (n-k)\Pr[A_{m,k}] = \frac{(n-k)N(N-1)\cdots(N-k+1)}{(2N-2)(2N-3)\cdots(2N-1-k)} \\ &\leq (n-k)\Big(\frac{N}{2N-2}\Big)^k \leq n\Big(\frac{1}{2-2/N}\Big)^k \leq n\Big(\frac{2}{3}\Big)^k,\end{aligned}$$

where the last inequality uses $n \geq 7$ and thus $N = (n+1)/2 \geq 4$. Hence for any constant $c > 2$, we know that

$$\Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq c\log_{3/2} n] \leq n\Big(\tfrac{2}{3}\Big)^{c\log_{3/2} n} = \tfrac{1}{n^{c-1}},$$

and

$$E[\mathrm{MEI}(P_{t+1,f_1})] = \sum_{k=0}^{+\infty} \Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq k] = \sum_{k=0}^{N} \Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq k]$$

$$= \sum_{k=0}^{c\log_{3/2} n - 1} \Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq k] + \sum_{k=c\log_{3/2} n - 1}^{N} \Pr[\mathrm{MEI}(P_{t+1}, f_1) \geq k]$$

$$\leq c\log_{3/2} n + (N - c\log_{3/2} n + 1) n \left(\tfrac{2}{3}\right)^{c\log_{3/2} n} \leq c\log_{3/2} n + o(1). \qquad \square$$

The example above showed that even in a perfectly symmetric situation, the NSGA-II with high probability selects a new parent population with high irregularities and relatively large areas on the Pareto front that are not covered by the population.

We now show that even more extreme examples can be constructed. We do not expect these to come up often in a regular run of the NSGA-II, but they underline that the drawbacks of working with the initial crowding distance can be tremendous. The constructed example is the combined population of the parent and offspring with the mutually different f_1 values for different individuals, and $f_1(R) = [0..\frac{1}{3}n + 1] \cup \{\frac{1}{3}n + 2i \mid i \in [1..\frac{1}{3}n]\}$. The the traditional survival selection will remove all individuals with f_1 value in $[1..\frac{1}{3}n + 1]$.

Lemma 6. *For all $n \in 3\mathbb{N}$, there is a combined parent and offspring population R such that $0^n, 1^n \in R$ and $\mathrm{MEI}(R, f_1) = 2$, but the population P' selected by the NSGA-II satisfies $\mathrm{MEI}(P', f_1) = \frac{1}{3}n + 2$.*

Proof. Let R be such that $f_1(R) = [0..\frac{1}{3}n + 1] \cup \{\frac{1}{3}n + 2i \mid i \in [1..\frac{1}{3}n]\}$ and that for any $x, x' \in R$, we have $f_1(x) \neq f_1(x')$. By construction, $0^n, 1^n \in R$ and $\mathrm{MEI}(R, f_1) = 2$. We note that exactly those $x \in R$ with $f_1(x) \in [1..\frac{1}{3}n + 1]$ have the smallest occurring crowding distance of $\frac{4}{n}$. Since these are $\frac{1}{3}n + 1$ elements of R and since $N = \frac{1}{2}|R| = \frac{1}{3}n + 1$ elements that have to be removed, they will all be removed in the selection step, leaving all points $(i, n - i)$ with $i \in [1..\frac{1}{3}n + 1]$ uncovered by the selected population. $\square$

The above two synthetic situations indicate the possible difficulties for the NSGA-II to approximate the Pareto front. We note here that in Section 6, our experiments shows the fair but not perfect approximation ability of the NSGA-II on the OneMinMax function, like for the problem size $n = 601$ and population size 76, at least 24 for the first quartile of MEI, while the ideal MEI is only 9. Although the above are not complete proofs, the drawbacks of the traditional survival selection indeed exist.

5 Updating Crowding Distance Can Help

As we have seen in the preceding section, the traditional way the NSGA-II selects the next parent population can lead to a not very evenly distributed parent

population. The proofs of Lemmas 5 and 6 also reveal a possible reason for this shortcoming: Since the crowding distance is computed only once and then used for all removals of individuals, it may happen that individuals are removed which, at the time of their removal, have a much larger crowding distance than at the start of the selection phase.

This phenomenon is heavily exploited in the construction of the very negative example in the proof of Lemma 6. In this example, the individuals x with $f_1(x) \in [1..\frac{1}{3}n]$ all have the smallest crowding distance of $4/n$, and thus are all removed in some arbitrary order. When the last individual is removed, its neighors on the front have objective values $(0, n)$ and $(\frac{1}{3}n + 1, n - (\frac{1}{3}n + 1))$. Consequently, this individual at the moment of its removal has a crowding distance of $2/3+2/n$ which is (for large n) much larger than its initial value of $4/n$, but also much larger than the crowding distance of $8/n$, which most of the remaining individuals still have. This example shows very clearly the downside of working with the initial crowding distance and at the same time suggests to work with the current crowding distance instead.

This is the road we will follow in this section. We first argue that there is an efficient implementation of the NSGA-II that repeatedly selects an individual with smallest crowding distance. We then show that this selection mechanism leads to much more balanced selections for the OneMinMax benchmark. We prove that the modified NSGA-II can achieve an MEI of at most $\frac{4n}{N-3}$, which is only by roughly a 4 factor larger than the ideal MEI of $(1+o(1))\frac{n}{N}$. Reaching such a balanced distribution is very efficient – once the two extremal points are found (in time $O(Nn\log(n))$), it only takes additional time $O(Nn)$ to find a population satisfying the MEI guarantee above. From this point on, the MEI never increases above $\frac{4n}{N-3}$.

5.1 Implementation of an NSGA-II algorithm Using the Current Crowding Distance for the Removal of Individuals

We first notice that the removal of one individual changes the crowding distance of at most four other individuals: The deleted individual has at most two neighbors in the sorted list of the front of population for each objective. Consequently, it suffices to update, that is, recompute, the crowding distance of these individuals. While this gives some optimism that an efficient implementation of the selection based on the current crowding distance exists, some more details have to be taken into account.

To describe them in sufficient clarity, let us assume that we have a set R of individuals which pairwise are not comparable (none dominates the other) or have

the same objective value. When optimizing OneMinMax, this set R will be the combined parent and offspring population; in the general case it will be the front F_{i^*}. Let us call $r = |R|$ the size of this set. Let us assume that we want to remove some number s of individuals from R, sequentially by repeatedly removing an individual with smallest current crowding distance, breaking ties randomly.

Besides keeping the crowding distance updated (which can be done in constant time per removal, as we just saw), we also need to be able to efficiently find and remove an individual with smallest (current) crowding distance, and moreover, a random one in case of ties. The detection and removal of an element with smallest key calls for a priority queue. Let us ignore for the moment the random tie-breaking and only discuss how to use a priority queue for the detection and removal of an individual with smallest crowding distance. We recall that a priority queue is a data structure that stored items together with a key, a numerical value assigned to each item. Standard priority queues support at least following three operations: Adding new items to the queue, removing from the queue an item with smallest key, and decreasing the key of an item in the queue. They do so with a time complexity that is only logarithmic in the current length of the queue.

For our problem, at the start of the selection phase, we add all individuals with their crowding distance as key to the priority queue. We repeatedly remove individuals according to the following scheme: (i) We find and remove from the priority queue an individual x with smallest key. We also remove x from R. (ii) We find the up to four neighbors of x in the two sortings of R according to the two objectives, compute their crowding distance, and update their keys in the priority queue accordingly. This is not a decrease-key operation (but an increase of the key), but such an increase-key can be simulated by first decreasing the key to an artificially small value (smaller than all real values that can occur, here for example -1), then removing the item with smallest key (which is just this item), and then adding it with the new key to the queue.

These two steps can be implemented in logarithmic time, since all operations of the priority queue take at most logarithmic time, except that we still need to provide an efficient way to find the neighbors of an element in the sortings. This is necessary to determine the up to four neighbors of x, but also to compute their crowding distance (which needs knowing their neighbors). To enable efficient computations of such neighbors, we use an additional data structure, namely for each objective a doubly-linked list that stores the current set R sorted according to this objective. This list data-structure must enable finding predecessors and successors (provided they exist) as well as the deletion of elements. Standard doubly-linked lists support these operations in constant time. We use this list in step (ii) above to find the desired neighbors. We also need to delete the removed individual x in step (i) from this list.

To allow finding individuals in the priority queue or the doubly-linked list, we need a helper data structure with pointers to the individuals in these data structures. This can be a simple array indexed by the initial set R.

With this setup, we can repeatedly remove an element with currently smallest crowding distance in logarithmic time. To add the random tie-breaking, it suffices to give each individual x a random second-priority key, e.g., a random number $r_x \in [0, 1]$. They key of an individual x used in the priority queue now is composed of the current crowding distance and this number r_x, where a key is smaller than another when its crowding distance is smaller or when the crowding distances are equal and the r_x number is smaller (lexicographic order of the two parts of the key). With these extended keys, the individuals with currently smallest crowding distance have the highest priority, and in case of several such individuals, the number r_x serves as random tie-breaker.

With this tie-breaking mechanism, we have now implemented a way to repeatedly remove an individual with smallest current crowding distance, breaking ties randomly, in time logarithmic in r, the size of the set R. Overall, thus our selection using the current crowding distance takes time at most $O(N \log N)$, which is small compared to the non-dominated sorting step, which takes quadratic time.

Without going into details, we note that when the possible crowding distance values are known and they are not too numerous, say there is an upper bound of S for their number, then using a bucket queue instead of a standard priority queue would give an complexity of order $O(S + N)$. This is slightly superior to the above runtime, e.g., for OneMinMax when $N = \Omega(n/\log n)$.

5.2 Runtime Analysis and Approximation Quality of the NSGA-II with Current Crowding Distance

We now conduct a mathematical analysis of the NSGA-II with survival selection based on the current crowding distance. The following lemma shows that invididuals with at least a certain crowding distance will certainly enter the next generation. The key argument in this proof is an averaging argument based on the observation that the sum of the crowding distances of all individuals other than the ones with infinite crowding distance is at most 4.

Lemma 7. *Let $N \geq 4$. Consider using the NSGA-II with survival selection based on the current crowding distance to optimize the n-dimensional OneMinMax function. Let t_0 be the first generation such that the two extreme points 0^n and 1^n are in P_{t_0}. Let $t \geq t_0$. Consider the selection of the next population P_{t+1} from R_t, which consists of N times removing an individual with smallest current crowding distance. Assume that at some stage of this removal process the individual x has*

Algorithm 3 NSGA-II with the survival selection using the current crowding distance

Uniformly at random generate the initial population $P_0 = \{x_1, x_2, \ldots, x_N\}$ for $x_i \in \{0,1\}^n, i = 1, 2, \ldots, N$.
for $t = 0, 1, 2, \ldots$ **do**
  Generate the offspring population Q_t with size N
  Using the fast-non-dominated-sort() in [DPAM02] to divide R_t into $F_1, F_2, \ldots$
  Find $i^* > 1$ such that $|\cup_{i=1}^{i^*-1} F_i| < N$ and $|\cup_{i=1}^{i^*} F_i| \geq N$, or $i^* = 1$ for $|F_1| \geq N$
  Using the crowding-distance() in Algorithm 1 to separately calculate the crowding distance of each individual in $F_1, \ldots, F_{i^*}$
  %% *Survival selection using the current crowding distance*
  while $|\cup_{i=1}^{i^*} F_i| \neq N$ **do**
    Let x be the individual with the smallest crowding distance in F_{i^*}, chosen at random in case of a tie
    Find four neighbors of x, two in the sorted list with respect to f_1 and two for f_2. Update the crowding distance of these four neighbors
    $F_{i^*} = F_{i^*} \setminus \{x\}$
  end while
  $P_{t+1} = \left(\cup_{i=1}^{i^*} F_i\right)$
end for

a crowding distance of $\frac{4}{N-3}$ *or higher. Then* $x \in P_{t+1}$*. Also,* P_{t+1} *surely contains the two extreme points.*

Proof. We consider first a single removal of an individual at some moment of the selection phase. Let R be the set of remaining individuals from the combined parent and offspring population. Note that $r := |R| > N$. Note also that, by definition of the crowding distance, there can be at most 4 individuals with infinite crowding distance. Since $N \geq 4$, such individuals are never removed, and consequently, R surely contains 0^n and 1^n. Let $y_1, \ldots, y_r$ be the sorting of R by increasing f_1 value and $z_1, \ldots, z_r$ be the sorting of R by increasing f_2 value that are used for the computation of the crowding distance. For $x \in R$, let $i, j \in [1..r]$ such that $x = y_i = z_j$ (we assume here that individuals have unique identifiers, so they are distinguishable also when having the same genotype; consequently, these i and j are uniquely defined). From the definition of $\mathrm{cDis}(x)$, we know the following. If $\{i, j\} \cap \{1, r\} \neq \emptyset$, then $\mathrm{cDis}(x) = \infty$. Otherwise, $\mathrm{cDis}(x) = (f_1(y_{i+1}) - f_1(y_{i-1}) + f_2(z_{j+1}) - f_2(z_{j-1}))/n$.

We compute

$$\begin{aligned}\sum_{i=2}^{r-1} f_1(y_{i+1}) - f_1(y_{i-1}) &= \sum_{i=2}^{r-1} f_1(y_{i+1}) - f_1(y_i) + f_1(y_i) - f_1(y_{i-1}) \\ &\leq 2\sum_{i=1}^{r-1} f_1(y_{i+1}) - f_1(y_i) \\ &= 2(f_1(y_r) - f_1(y_1)) = 2(n-0) = 2n,\end{aligned}$$

the latter by our insight that R contains the two extremal individuals. An analogous estimate holds for f_2 and the z_j. This allows the estimate

$$\sum_{x \in R^*} n \,\mathrm{cDis}(x) \leq \sum_{i=1}^{r-1} f_1(y_{i+1}) - f_1(y_{i-1}) + \sum_{j=1}^{r-1} f_2(z_{j+1}) - f_2(z_{j-1}) \leq 4n,$$

where we write R^* for the individuals in R having a finite crowding distance. Since $|R^*| \geq r-4$, a simple averaging argument shows that there is an $x \in R^*$ with $\mathrm{cDis}(x) \leq \frac{4}{r-4}$. Hence also the individual that is removed has a crowding distance of at most $\frac{4}{r-4} \leq \frac{4}{N-3}$.

Now looking at all removals in this selection step, we see that never an individual with crowding distance above $\frac{4}{N-3}$ is removed. □

The result just shown implies that no large empty interval is created by the removal of a solution from R_t in the selection phase. Slightly more involved arguments are necessary to argue that the MEI-value decreases relatively fast. It is not too difficult to see that if the set of f_1 values in P_t contains a large empty interval (the same will then be true for f_2), then this interval can be reduced in length by at least one via the event that one of the individuals corresponding to the boundaries of the empty interval create an offspring "inside the interval". What needs more care is that we require such arguments for all large empty intervals in parallel. For this, we regard how an empty interval shrinks over a longer time. Since this shrinking is composed of independent small steps, we can use strong concentration arguments to show that an interval shrinks to the desired value in the desired time with probability at least $1 - \exp(-\Omega(n))$. This admits a union bound argument to extend this result to all empty intervals.

A slight technical challenge is to make precise what "one interval" means over the course of time (recall that in one iteration, we generate N offspring and then remove N individuals from R_t). We overcome this by regarding a half-integral point $i + 0.5$ and the corresponding interval

$$I_i = [\max\{f_1(x) \mid x \in P_t, f_1(x) \leq i + 0.5\} .. \min\{f_1(x) \mid x \in P_t, f_1(x) \geq i + 0.5\}].$$

This definition is unambiguous. It gives room to some strange effects, which seem hard to avoid, nevertheless. Assume that $I_i = [i..b]$ for some b sufficiently larger than i. Assume that there is a single individual x with $f_1(x) = i$ in P_t. Assume that x has an offspring y with $f_1(y) = i + 1$. If both x and y survive into the next generation, then I_i has suddenly shrunk a lot to $[i..i + 1]$. If y survives but x does not, then I_i has changed considerably to $[a..i + 1]$, where $a = \max\{f_1(x) \mid x \in P_t, f_1(x) \leq i\}$. Since in each iteration N individuals are newly generated and then N are removed, we do not see a way to define the empty intervals in a more stable manner. Nevertheless, our proof below will be such that it covers all these cases in a relatively uniform manner.

We recall from Section 2 that *fair selection* means that each individual of the parent population generates exactly one offspring and *random selection* means that N times an individual is chosen uniformly at random from the parent population to generate one offspring.

Lemma 8. *Let $N \geq 4$. Consider using the NSGA-II with fair or random parent selection, with one-bit mutation, and with survival selection using the current crowding distance, to optimize the n-dimensional* OneMinMax *function. Let t_0 be the first generation that the two extreme points 0^n and 1^n are contained in P_{t_0}. Let $t_1 \geq t_0$ be the first generation such that $\mathrm{MEI}(P_{t_1}, f_1) \leq \max\{\frac{4n}{N-3}, 1\} =: L$. Then $t_1 - t_0 = O(n)$, both in expectation and with probability $1 - o(1)$. Also, for all $t > t_1$, we have $\mathrm{MEI}(P_t, f_1) \leq L$ with probability one.*

Proof. Let $i \in [0..n-1]$. Let $t \geq t_0$ and X_t be the length of the empty interval in $f_1(P_t)$ containing $i + 0.5$, that is,

$$X_t = \min\{f_1(x) \mid x \in P_t, f_1(x) \geq i + 0.5\} - \max\{f_1(x) \mid x \in P_t, f_1(x) \leq i + 0.5\}.$$

We first prove that once $X_t \leq L$, we have $X_{t'} \leq L$ for all $t' \geq t$ with probability 1. Let X'_t be the length of the empty interval in $f_1(R_t)$ that contains $i + 0.5$. Since $X_t \leq L$ and $R_t \supseteq P_t$, we have $X'_t \leq X_t \leq L$. We prove now the slightly stronger statement that once $X'_t \leq L$, we have $X_{t'} \leq L$ for all $t' \geq t$. It suffices to regard one iteration, so assume by way of contradiction that $X_{t+1} > L$. Note that P_{t+1} is obtained from R_t by repeatedly removing an individual with smallest crowding distance. Consider the first removal step after which the empty interval containing $i + 0.5$ is larger than L. Let $[a..b]$ be the empty interval containing $i + 0.5$ before the removal. To increase the size of this empty interval, one of a and b must be removed. Assume, without loss of generality, that this was a. Removing a, by definition of the crowding distance, creates an empty interval of length at most n times the current crowding distance of a. By Lemma 7, this current crowding distance is at most $\frac{4}{N-3}$, contradicting our assumption that the removal of a creates an empty interval of length larger than L.

We now consider the situation that $X_t > L$. Let a and b with $a \leq b$ be the two ends of the interval in $f_1(P_t)$ that contains $i + 0.5$. Let $x_a, x_b \in P_t$ respectively be individuals with f_1 value a and b. When using fair parent selection, the probability to create an offspring with f_1 value in $\{a+1, b-1\}$ is at least

$$1 - \frac{a}{n}\frac{n-b}{n} \geq 1 - \left(\frac{a+n-b}{2n}\right)^2 \geq \frac{3}{4} =: p.$$

For random parent selection, noting that one of $\frac{n-a}{n}$ and $\frac{b}{n}$ is at least $\frac{1}{2}$, this probability is at least $\frac{1}{2}(1-(1-1/N)^N) \geq \frac{1}{2}(1-1/e) =: p$.

Let us assume that this positive event has happened, that is, at least one individual with f_1 value in $\{a+1, b-1\}$ was created. Then the interval in $f_1(R_t)$ containing $i + 0.5$ has length $X'_t \leq X_t - 1$ (note that, depending on the value of i and the outcome of the offspring generation, this interval can be any of $[a..a+1]$, $[a..b-1]$, $[a+1..b-1]$, $[a+1..b]$, and $[b-1..b]$, but all arguments below hold for any of these cases). By our earlier argument, we now have $X_{t+1} \leq X_t - 1$ with probability one. This argument shows that in any iteration starting with $X_t > L$, regardless of what happened in previous iterations, we have a probability of at least p of reducing X_t by at least one.

Now we consider the probability of the event A that X_t drops below L in $T = \lceil 2(1/p)n \rceil$ generations. We consider a random variable $Y = \sum_{i=1}^{T} Y_i$ where $Y_1, \ldots, Y_T$ are independent Bernoulli random variables with success probability of p. Then $X_{t_0} - X_{t_0+T} = \sum_{t=t_0}^{T-1}(X_t - X_{t+1})$ stochastically dominates $Z := \min\{Y, X_{t_0} - L\}$. Since $E[Y] \geq 2n$, applying the multiplicative Chernoff bound (see, e.g., [Doe20, Theorem 1.10.7]), we have

$$\Pr[Y \leq X_{t_0} - L] \leq \Pr[Y \leq n - L] \leq \Pr[Y \leq n] \leq \exp(-n/2).$$

Hence

$$\Pr[A] \geq \Pr[X_{t_0} - X_{t_0+T} \geq X_{t_0} - L] \geq \Pr[Z \geq X_{t_0} - L] \geq 1 - \exp(-n/2).$$

Together with the first part in this proof, we know that once A happens, for any generation $t' \geq t_0 + T$, we have $X_{t'} \leq L$.

Note that the above discussed X_t is corresponding to one given $i \in [0..n-1]$. A union bound over all $i \in [0..n-1]$ gives that for any generation after the (t_0+T)-th generation, with probability at least

$$1 - n\exp(-n/2)$$

all empty intervals in the population will have the length at most L, that is, we have $\mathrm{MEI}(P_t, f_1) \leq L$ for all $t \geq t_0 + T$. This proves the claimed bound with high probability.

For the claimed bound on the expectation, we note that we can repeat our argument above since we did not make any particular assumptions on the initial state. Consequently, the probability that $\mathrm{MEI}(P_{t_0+\lambda T}, f_1) > L$ is at most $(n\exp(-n/2))^\lambda$. This immediately implies that that the expected time to have all empty intervals of length at most L is at most $t_0 + O(1)T = t_0 + O(n)$. □

It remains to analyze the time it takes to have the two extreme points 0^n and 1^n in the population. This analysis is very similar to the analysis of the original NSGA-II with population size large enough that the full Pareto front is found in [ZLD22, Theorems 2 and 6]. Since the proof below also applies to the original NSGA-II, we formulate the following result for both algorithms.

Lemma 9. *Consider using the NSGA-II (Algorithm 2 or 3) with one of the following six ways to generate the offspring, namely, applying fair selection, random selection, or binary tournament selection and applying one-bit mutation or standard bit-wise mutation. Then after an expected number of $O(n\log n)$ iterations, that is, an expected number of $O(Nn\log n)$ fitness evaluations, the two extreme points 0^n and 1^n are contained in the population.*

Proof. We first consider the time to generate 1^n, that is, the unique search point with largest f_1 value. Let $t \geq 0$. Let $x_{\max}$ be an individual in the parent population P_t with largest f_1 value and with infinite crowding distance. Let $k = f_1(x_{\max})$. Let p_s^k denote the probability that $x_{\max}$ appears at least once in the N individuals selected to generate offspring and let p_+^k denote the probability of generating an individual with larger f_1 value from $x_{\max}$ via the mutation operator. Note that if there exists individuals with f_1 value larger than $f_1(x)$, then one such individual will survive to P_{t+1}, and P_{t+1} will have an individual with f_1 fitness at least $f_1(x_{\max}) + 1$. The expected number of iterations until this happens is at most $1/(p_s^k p_+^k)$. It is not difficult to see that the largest f_1 value in P_t cannot decrease. Since $f_1(1^n) = n$, the expected number of iterations to reach 1^n is at most $\sum_{k=0}^{n-1} \frac{1}{p_s^k p_+^k}$.

It is not difficult to see that $p_s^k = 1$ for the fair selection and at least $1 - (1-1/N)^N \geq 1 - 1/e$ for random selection. For binary tournament selection, there are at most two individuals with infinite crowding distance but f_1 value different from $f_1(x_{\max})$. Hence $p_s^k \geq 1 - \left(1 - \frac{1}{N}\frac{N-2}{N}\right)^N = 1 - \exp(-(N-2)/N) = \Theta(1)$ in this case. It is also not difficult to see that $p_+^k = (n-k)/n$ for the one-bit mutation, and $p_+^k \geq \frac{n-k}{n}(1-\frac{1}{n})^{n-1} \geq \frac{n-k}{en}$. Hence the expected number of iterations to find 1^n is at most

$$\sum_{k=0}^{n-1} \frac{1}{p_s^k p_+^k} = O\left(\sum_{k=0}^{n-1} \frac{n}{n-k}\right) = O(n\log n).$$

Similarly, we could show that the expected number of iterations to have 0^n in the population is also $O(n \log n)$. Hence, the expected number of iteration to have the two extreme points in the population is $O(n \log n)$. Since each iteration uses N fitness evaluations, the expected number of fitness evaluations is $O(Nn \log n)$. □

From Lemmas 8 and 9, we easily obtain the following theorem on the approximation ability of the NSGA-II using the current crowding distance.

Theorem 10. *Let $N \geq 4$. Consider using the NSGA-II with fair or random parent selection, with survival selection using the current crowding distance, and one-bit mutation to optimize the n-dimensional* OneMinMax *function. Then after an expected number of $O(Nn \log n)$ fitness evaluations, a population containing the two extreme points 0^n and 1^n and with $\mathrm{MEI}(P_t, f_1) \leq \frac{4n}{N-3}$ is reached and kept for all future time.*

Recalling from Lemma 3 that the ideal maximal empty interval size is $\frac{n}{N-1}$, Theorem 10 shows that the gaps in the Pareto front are at most by around a factor of 4 larger than this theoretical minimum. Also, comparing the runtimes in Lemma 8 with those in Lemma 9 or Theorem 10, we see that the cost for reaching a good approximation is asymptotically negligible compared to the one proven for reaching the two extreme points.

6 Experiments

In Section 4 we only conducted a theoretical analysis of two synthetic situations to show the risk that the traditional NSGA-II has difficulties in approximating the Pareto front since the complicated population dynamics prevented us from a complete mathematical analysis of this question. Hence, it remains to experimentally evaluate the approximation ability of this algorithm. In Section 5, we proved that the modified NSGA-II leaves gaps on the Pareto front that are asymptotically at most a factor of 4 larger than those given by a perfect approximation of the Pareto front. Since we do not prove a matching lower bound, it is again necessary to experimentally investigate how larges these gaps are in actual runs. This section will focus on the above two aspects.

6.1 Settings

To investigate the questions above, we conduct experiments with the following settings.

- Problem: OneMinMax. OneMinMax is the analyzed benchmark in our theoretical results in Sections 4 and 5.

- Problem size n: 601. Given that the OneMinMax problem is an easy multi-objective problem, this is a moderate problem size. Such a choice is sensible, since on a too small size we will not gain reliable insights, whereas insights obtained on a large problem size raise the question whether they still apply to practically relevant problem sizes. We use the odd number $n = 601$ to include the setting discussed in Lemma 5.

- Algorithms: We regard the classic NSGA-II using the initial crowding distance for the selection and our variant, using the current crowding distance. As in the first runtime analysis of the NSGA-II [ZLD22], we do not use crossover. That is, the mutation is the only operator to generate the offspring population.

- Mating selection and mutation strategy: fair selection and one-bit mutation. These two are discussed in the runtime for the NSGA-II with the survival selection using the current crowding distance, see Theorem 10. They are enough to verify our findings about the approximation difficulty of the traditional NSGA-II and improved approximation ability for the modified one with the survival selection using the current crowding distance. We believe similar behaviors for other popular strategies.

- Population size N: $(n+1)/2 = 301$, $\lceil (n+1)/4 \rceil = 151$, and $\lceil (n+1)/8 \rceil = 76$. We choose $(n+1)/2$ as this value is used in Lemma 5. We did not regard larger population sizes, since for $N = n + 1$ experiments where conducted in [ZLD22]. Other two smaller population sizes are used to see a more general approximation ability of the NSGA-II.

- 20 independent runs for each setting.

6.2 Results

Our focus in our experiments is the maximal length of an empty interval on the Pareto front, that is, the MEI value defined earlier. As long as the population has not fully spread out on the Pareto front, that is, the extremal solutions 0^n and 1^n are not yet part of the population, enlarging the spread of the population is more critical than a balanced distribution in the known part of the front. For this reason, we only regard times after both extremal solutions have entered the population.

To see whether the approximation quality changes over time, we regard separately the two intervals of [1..100] and [3001..3100] generations after finding the two extremal solutions. We collect statistical data on the MEI value in these intervals in Table 1 and we display exemplary runs in Figure 1.

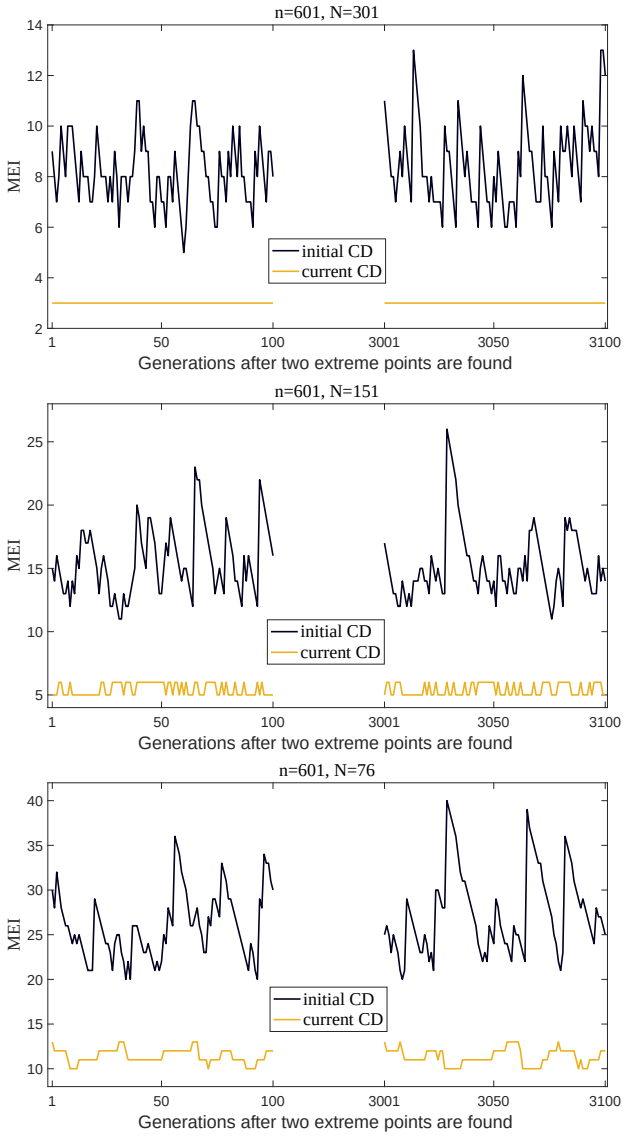

Figure 1: The maximal empty interval sizes MEI for [1..100] and [3001..3100] generations after two extreme points are reached in one exemplary run.

Table 1: The first, second, and third quartiles (collected in the format of $(\cdot,\cdot,\cdot)$) for the maximal empty interval sizes MEI within [1..100] and [3001..3100] generations after the two extremal points have entered the population.

Generations	[1..100]	[3001..3100]
$N = 301$		
Initial CD	(7,8,9)	(7,8,9)
Current CD	(3,3,3)	(3,3,3)
$N = 151$		
Initial CD	(14,15,17)	(13,14,16)
Current CD	(5,5,6)	(5,5,6)
$N = 76$		
Initial CD	(25,27.5,30)	(24,26,30)
Current CD	(11,11,12)	(11,11,12)

The ideal MEI value $\lceil n/(N-1) \rceil$ for OneMinMax with $n = 601$ and population size $N = 301, 151, 76$ is $3, 5$, and 9, respectively. From Table 1, we see that the modified NSGA-II with the survival selection using the current crowding distance can reach the ideal MEI for $N = 301$ and 151 and is slightly above the ideal value for $N = 76$. In contrast, the traditional NSGA-II shows median MEI values of $8, 14, 26$ in the later (better) time interval. This is more than twice the ideal MEI and the median values of our variant of the NSGA-II. We observe minimally worse values for the classic NSGA-II in the first time interval (right after the extremal points were found), but the difference is small and rather suggests that also before the extremal points are found, the population is evenly spread out (in the part of the Pareto front explored so far).

In Figure 1, we see that the MEI value oscillates considerably for the classic NSGA-II, whereas it is relatively stable for the NSGA-II using the current crowding distance for the selection. This appears to be a second advantage of our new variant of the NSGA-II.

Our experimental data is not sufficient to answer the question if the traditional NSGA-II suffers from super-constant MEI values. Our theoretical result in Lemma 5 could be seen as an indication that logarithmic MEI values can show up (or MEI values of order $\Theta(\frac{n}{N} \log N)$ for general values of $N \leq n$). To answer this question, significantly more experiments with truly large problem sizes would be necessary (due to the slow growth behavior of logarithmic functions). For our purposes, our results, however, are fully sufficient. They show clearly that the version of the NSGA-II proposed in this work leads to much smaller and more stable MEI values. Not surprisingly, the experimentally observed MEI values for the new

algorithm variant are much better than the mathematical guarantee given in Theorem 10. Explaining this discrepancy, most likely by sharpening the mathematical guarantees, is another interesting problem for future research.

7 Conclusion

The so far only previous runtime analysis of the NSGA-II [ZLD22] did not regard the case that the population size is smaller than the size of the Pareto front. In this work, we regard this situation and discuss how well the population evolved by the NSGA-II approximates the Pareto front. Our theoretical analysis of two artificial cases and our experiments give a mixed picture. However, they also suggest that the reason for the not fully satisfying approximation behavior is the fact that the selection of the next parent population is based on the initial crowding distance of individuals in the combined parent and offspring population, which can be very different from the crowding distance at the moment when an individual is removed. This may lead to large uncovered areas on the Pareto front. We therefore propose to build the selection on the current crowding distance. We show that this selection strategy can be implemented efficiently. We then prove that this selection leads to gaps in the Pareto front of OneMinMax that are only a constant factor larger than in an ideal solution. Our experiments confirm the superiority of the new selection strategy.

From our proofs, we conjecture that similar results can be obtained for other classic benchmark problems such as LOTZ or the large front problem [HN08]. We are also optimistic that our results and methods can give interesting results for other approximation measures such as the hypervolume and the inverted generational distance.

References

[AD11] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics*. World Scientific Publishing, 2011.

[BFN08] Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature, PPSN 2008*, pages 651–660. Springer, 2008.

[BFQY20] Chao Bian, Chao Feng, Chao Qian, and Yang Yu. An efficient evolutionary algorithm for subset selection with general cost constraints. In *AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 3267–3274. AAAI, 2020.

[BQT18] Chao Bian, Chao Qian, and Ke Tang. A general approach to running time analysis of multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1405–1411. IJCAI, 2018.

[BT03] Peter AN Bosman and Dirk Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(2):174–188, 2003.

[Cra21] Victoria G. Crawford. Faster guarantees of evolutionary algorithms for maximization of monotone submodular functions. In *International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 1661–1667. ijcai.org, 2021.

[DN20] Benjamin Doerr and Frank Neumann, editors. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer, 2020.

[Doe20] Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 1–87. Springer, 2020. Also available at `https://arxiv.org/abs/1801.06733`.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[DZ21] Benjamin Doerr and Weijie Zheng. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 12293–12301. AAAI Press, 2021.

[Gie03] Oliver Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *IEEE Congress on Evolutionary Computation, CEC 2003*, volume 3, pages 1918–1925. IEEE, 2003.

[GL10] Oliver Giel and Per Kristian Lehre. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18:335–356, 2010.

[HN08] Christian Horoba and Frank Neumann. Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization.

In *Genetic and Evolutionary Computation Conference, GECCO 2008*, pages 641–648. ACM, 2008.

[HZ20] Zhengxin Huang and Yuren Zhou. Runtime analysis of somatic contiguous hypermutation operators in MOEA/D framework. In *AAAI Conference on Artificial Intelligence, AAAI 2020*, volume 34, pages 2359–2366. AAAI, 2020.

[HZCH19] Zhengxin Huang, Yuren Zhou, Zefeng Chen, and Xiaoyu He. Running time analysis of MOEA/D with crossover on discrete optimization problem. In *AAAI Conference on Artificial Intelligence, AAAI 2019*, volume 33, pages 2296–2303. AAAI, 2019.

[Jan13] Thomas Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer, 2013.

[LLTY15] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 48(1):1–35, 2015.

[LTDZ02] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10:263–282, 2002.

[LTZ+02] Marco Laumanns, Lothar Thiele, Eckart Zitzler, Emo Welzl, and Kalyanmoy Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *International Conference on Parallel Problem Solving from Nature, PPSN 2002*, pages 44–53. Springer, 2002.

[LZZZ16] Yuan-Long Li, Yu-Ren Zhou, Zhi-Hui Zhan, and Jun Zhang. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20:563–576, 2016.

[NW10] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.

[QBF20] Chao Qian, Chao Bian, and Chao Feng. Subset selection by pareto optimization with recombination. In *AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 2408–2415. AAAI, 2020.

[QYT+19] Chao Qian, Yang Yu, Ke Tang, Xin Yao, and Zhi-Hua Zhou. Maximizing submodular or monotone approximately submodular functions by multi-objective evolutionary algorithms. *Artificial Intelligence*, 275:279–294, 2019.

[RNNF19] Vahid Roostapour, Aneta Neumann, Frank Neumann, and Tobias Friedrich. Pareto optimization for subset selection with dynamic cost constraints. In *AAAI Conference on Artificial Intelligence, AAAI 2019*, volume 33, pages 2354–2361. AAAI, 2019.

[ZLD22] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the non-dominated sorting genetic algorithm ii (NSGA-II). In *AAAI Conference on Artificial Intelligence, AAAI 2022*. AAAI Press, 2022. To appear. See arXiv version in https://arxiv.org/abs/2112.08581.

[ZQL+11] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1:32–49, 2011.

[ZT98] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Parallel Problem Solving from Nature, PPSN 1998*, pages 292–301. Springer, 1998.

[ZYQ19] Zhi-Hua Zhou, Yang Yu, and Chao Qian. *Evolutionary learning: Advances in theories and algorithms*. Springer, 2019.